02 3D 게임 프로그래밍을 위한 기초 수학 - 2. 벡터(덧셈)

3D 게임 프로그래밍을 위한 기초 수학에 대한 연재 강좌를 시작합니다.

저도 수학에 대해서 잘 모르지만 공부하면서 알게된 지식을 공유하기 위한 목적으로 올립니다.

벡터와 삼각함수 부분을 연재해 나갈 계획이며,

수학에 기초가 약한 분들을 대상으로 쉽게 쓰려고 합니다.

따라서 이미 베테랑이신 분들은 다 아는 내용일지도 모릅니다.

혹시 내용중 잘못된 부분에 대한 지적이나 의견을 주시면 너무 감사하겠습니다.

적극적인 피드백 부탁드립니다.^^

지난 강좌

[3D 게임 프로그래밍을 위한 기초 수학 - 1. 시작](http://lab.gamecodi.com/board/zboard.php?id=GAMECODILAB_Lecture_series&no=121&z=)

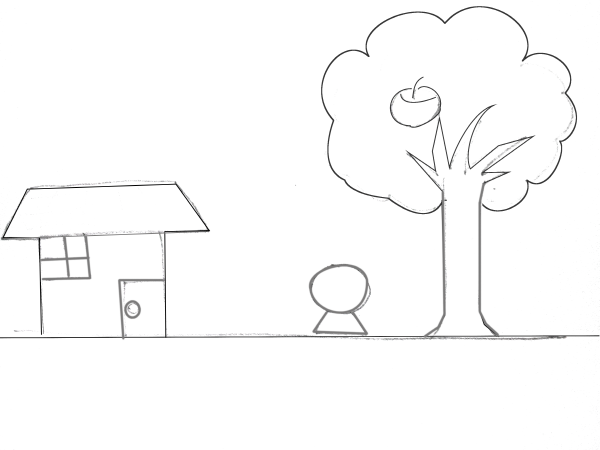
**벡터의 정의**

크기와 방향을 동시에 나타내는 물리량을 말한다.

(네이버 백과사전. <http://terms.naver.com/entry.nhn?docId=940915&cid=47338&categoryId=47338)>

벡터라는 단어를 처음 접한다면 위 내용이 무슨 뜻인지 알 수가 없습니다.

벡터의 사전적 정의는 잠시 잊어 버리고 그림을 보면서 알아보도록 하죠.



집과 나무, 사람이 있는 그림입니다.

여기서 한 가지 질문을 드리겠습니다. 저 그림에서 사람이 어디에 위치해 있죠?

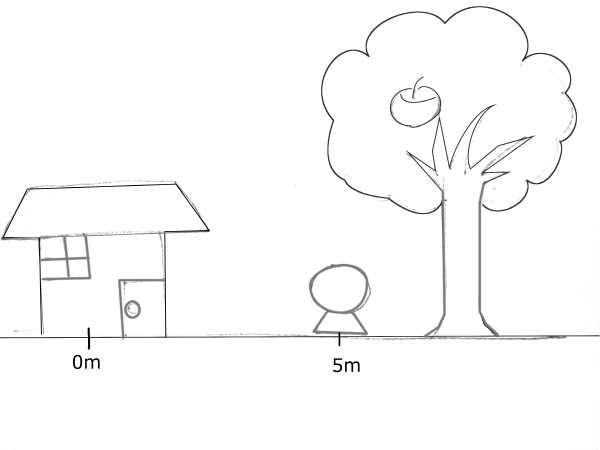
단순히 그림만 본다면 이렇게 대답할 것입니다.

“집에서 조금 떨어져 있고 나무 아래에 있는데요?”

그렇다면 이번에는 조금 더 수학적으로 접근해 보도록 합시다.

아래 그림을 보고 다시 얘기해 주세요.

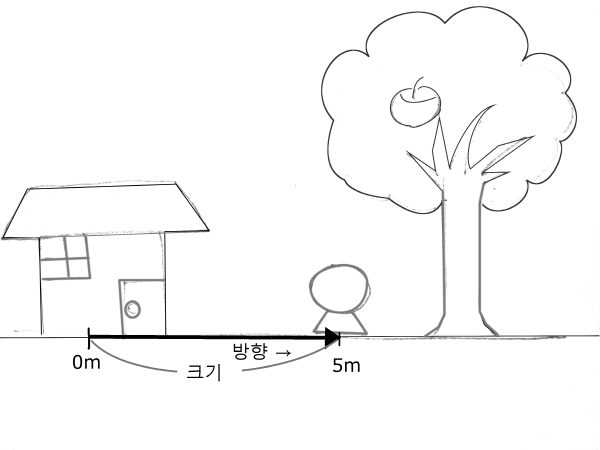
사람이 집에서 얼마만큼 떨어져 있나요?



이제는 이렇게 대답할 수 있겠죠.

“집에서부터 5m 떨어져 있네요.”

네. 맞습니다. 그럼 방금 답하신 내용을 문장이 아닌 벡터로 표현해 보도록 하겠습니다.



위 그림은 사람의 위치를 문장이 아닌 벡터로 나타낸 것입니다.

벡터는 크기와 방향을 갖는다고 백과사전에 나와 있었죠?

저 그림에서 사람을 가리키는 벡터는 크기가 5이며, 방향은 화살표가 가리키는 오른쪽 방향이다 라고 이해하시면 됩니다.

여기서 크기라는 것은 화살표의 길이를 의미 합니다.

그렇다면 도대체 왜 벡터라는 것을 쓰는 건가요?

위치를 나타내려면 x,y처럼 점으로 표현할 수 도 있지 않나요?

맞습니다. 점으로도 위치를 표현할 수 있죠.

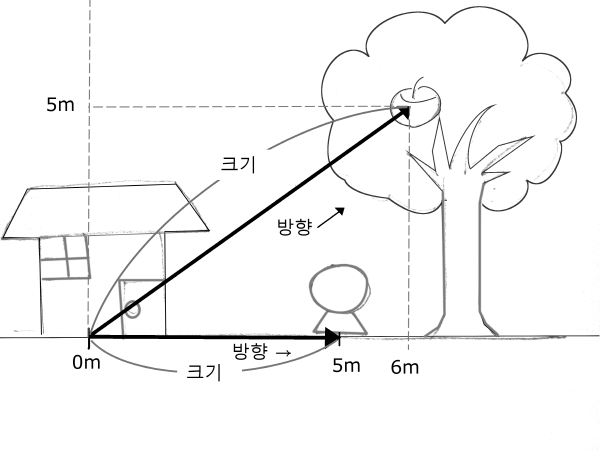
하지만 벡터를 사용하는 이유는 벡터의 특성들을 게임에 활용하기 위해서입니다.

점으로만 사용했을 때 보다 훨씬 더 다양한 모습들을 쉽게 계산할 수 있거든요.

앞으로 그 특성들을 하나하나 알려드릴 텐데요,

그 전에 벡터 하나를 더 추가해보도록 하겠습니다.

그림에서 사과가 보이시나요? 사과를 가리키는 벡터를 그려볼까요?



사람의 위치를 나타내는 벡터와 마찬가지로

집에서부터 사과의 위치까지 화살표를 그리면 사과를 가리키는 벡터가 되죠.

크기는 화살표의 길이, 방향은 집에서부터 사과를 향하고 있습니다.

게임 안에 존재하는 어떤 물체라도 벡터를 통해서 그 위치를 표현할 수 있습니다.

심지어 눈에 보이지 않는 속도, 힘 등을 표현하는 데에도 벡터가 사용됩니다.

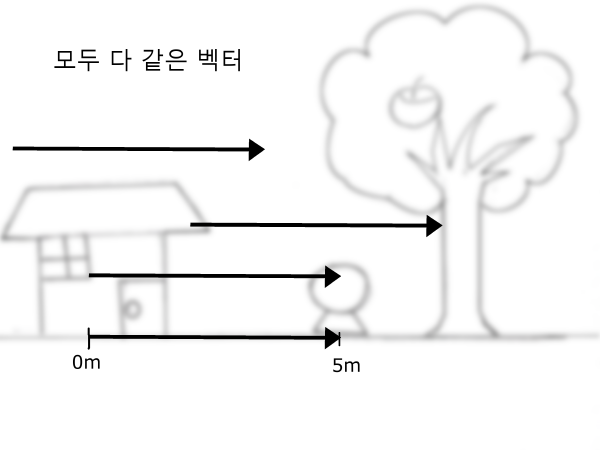
그럼 이제 벡터의 특성에 대해서 알아보도록 하겠습니다.

위 그림에서 벡터의 화살표가 가리키는 위치는 점으로 표현했을 때의 위치와 동일하죠.

따라서 벡터와 점이 무슨 차이가 있는지 모를 겁니다.

하지만 벡터는 크기와 방향으로 구성된다고 말했습니다.

즉, 크기와 방향이 같다면 어디에 있는 벡터든지 모두 같은 벡터로 취급할 수 있다는 뜻입니다.



위 그림에서 네 개의 벡터는 각각 위치가 다르지만 모두 다 같은 벡터입니다.

“시작 위치가 다르고 화살표의 끝이 가리키는 곳이 다른데 어떻게 같은 벡터가 될 수 있나요?”

벡터의 뜻을 찾아봤을 때 크기와 방향으로 이루어져 있다고 말씀드렸습니다.

시작 위치나 끝 위치 등의 개념은 갖고 있지 않죠.

다만 게임 내에서는 원점(0,0)을 시작으로 각각의 객체의 위치까지 벡터로 이은다고 생각하면

벡터를 이용하여 좌표를 나타낼 수 있으므로 여러 가지 편리한 점이 많습니다.

크기와 방향이 같으면 모두 같은 벡터이다.

그렇다면 벡터는 시작점에 고정되어 있는 것이 아니다 라고 이해하면 되는 건가요?

그렇습니다. 벡터는 어느 한 점에 고정되어 있지 않습니다.

따라서 크기와 방향이 같다면 이리저리 옮겨 다녀도 모두 같은 벡터라는 것이죠.

같은 벡터이기 때문에 결국 벡터 연산에 있어서 크기와 방향만 같다면

어디에 붙여놔도 동일한 계산 결과가 나온다는 뜻입니다.

벡터를 이해하려면 약간의 상상력을 더해야 합니다.

눈에 보이지는 않지만 우리 주변에 무수히 많은 벡터가 있다고 생각해 봅시다.

내 오른쪽 1m 옆에 누군가가 서 있다면 나로부터 오른쪽으로 1m 길이를 갖는

화살표(벡터)가 존재한다고 상상해볼 수 있습니다.

또한 땅으로부터 내 머리 끝까지 화살표로 잇는다면 내 키를 표현하는 벡터가 생기게 됩니다.

내가 자리에 앉거나 바닥에 눕는다면 그 벡터의 길이와 방향도 같이 변하게 되겠죠.

대부분의 게임 프로그래밍에서는 캐릭터를 이동시킬 때 벡터의 연산을 이용합니다.

따라서 벡터의 덧셈, 뺄셈 등의 연산 방법을 배우는 것은

게임 프로그래밍을 하는데 가장 기초적인 수학적 기반을 쌓는 셈이죠.

혹, 벡터라는 것을 모르고 게임 프로그래밍을 하고 있더라도 걱정할 필요는 없습니다.

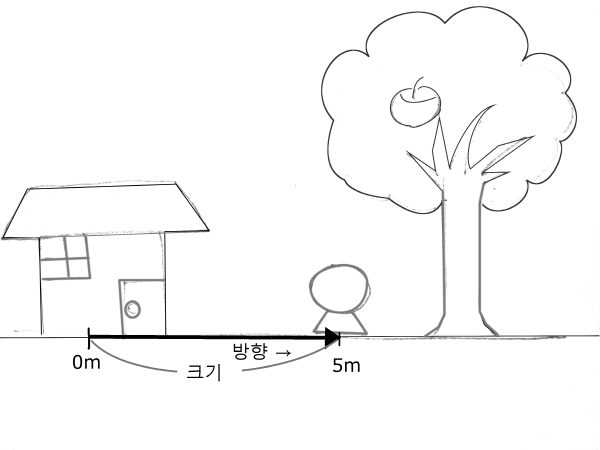
용어의 차이만 있을 뿐이지 코드 속에서는 이미 벡터의 개념을 활용하고 있을 것이기 때문이지요.

만약 유니티 엔진을 사용하여 게임을 개발하고 있다면 씬에 빈 오브젝트를 하나 올려 놓는 것 만으로도

벡터를 사용하게 되는 셈입니다.(Transform의 position값이 벡터로 되어있기 때문입니다.)

이런 내용을 머릿속에 담고 이제부터 벡터끼리 계산하는 방법을 알아보도록 하겠습니다.

**벡터의 덧셈**



그림속의 사람이 게임 캐릭터라고 생각해 봅시다.

현재 집으로부터 5m 위치에 서 있는데 떨어지는 사과를 받기 위해 6m 지점으로 이동시키려고 합니다.

벡터를 사용한다면 이걸 어떻게 표현할 수 있을까요?

현재 5m 지점에 서 있고, 사과는 그보다 1m 오른쪽에 있는 6m 지점으로 떨어질 테니까 현재 위치에서 오른쪽으로 1m 이동하면 되겠네요.

이것을 벡터로 표현한다면 a = [5,0] 이 되겠군요. [x좌표, y좌표] 순서로 씁니다.

그리고 사과가 떨어질 위치는 집으로부터 오른쪽으로 6m 지점에 위치해 있으니

c = [6,0] 이라고 표현할 수 있습니다.

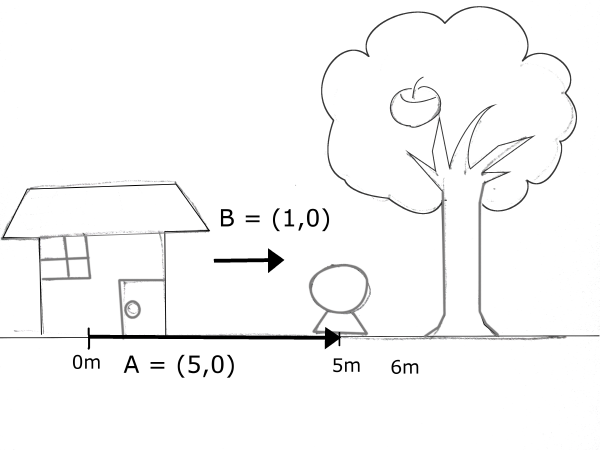
(캐릭터를 A, 사과가 떨어질 위치를 C라고 두겠습니다)

또한 현재 위치에서 오른쪽으로 1m 이동하면 사과가 떨어질 위치까지 갈 수 있으니

이것도 벡터로 표현한다면 b = [1,0] 이 되겠네요.

이렇게 세 개의 벡터가 생겨났습니다.

이걸 그림으로 그려 볼까요?



우선 벡터 a와 벡터 b를 그려봤습니다.

앗, 그런데 벡터 b는 왜 저 위에 있지요?

b = [1,0] 이라면 집의 위치에서부터 시작해서 오른쪽으로 1m 지점에 위치한 벡터로 그려야 맞지 않나요?

벡터는 크기와 방향이 같다면 어디에 있든지 같은 벡터라고 말씀 드렸었죠?

어떠한 특정한 지점에 고정되어 있지 않다는 것을 기억해 두세요.

이 특성은 벡터의 덧셈을 할 때 아주 유용하게 사용됩니다.

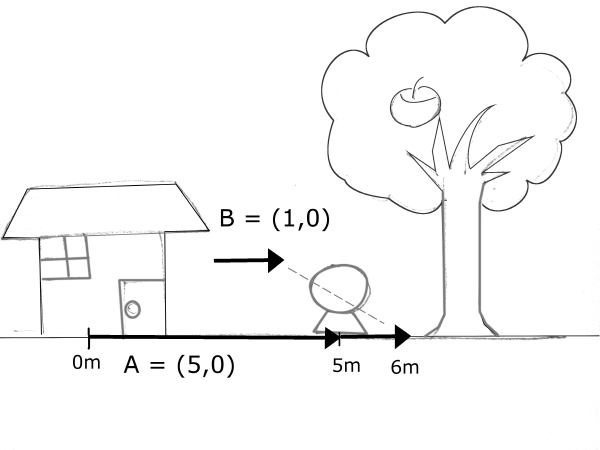
벡터의 덧셈은 아주 간단한데요 꼬리에서 머리로 이어주면 됩니다.

꼬리는 벡터가 시작되는 부분을 말하고요,

머리는 벡터의 화살표 부분을 말합니다.

두 벡터의 덧셈은 한 벡터의 꼬리에서부터 다른 벡터의 머리까지 쭉 이어줬을 때 생기는 새로운 벡터를 의미합니다.

먼저 벡터 b를 움직여서 벡터 a와 서로 붙도록 만들어 봅시다.

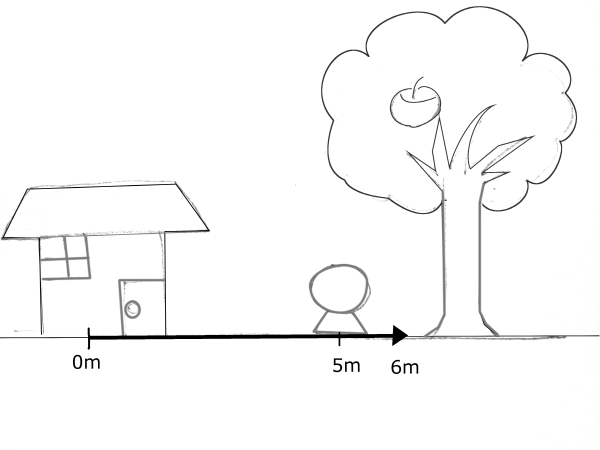


이제 이렇게 만든 벡터를 아까 말씀드린 덧셈 방법에 따라 더해 봅시다.

덧셈은 꼬리에서 머리로 이어주면 된다고 했습니다.

벡터 a의 꼬리가 가리키는 지점에서 벡터 b의 머리가 가리키는 지점 까지 이어주면

다음과 같이 새로운 벡터를 얻을 수 있습니다.



새로 생긴 벡터 c는 결국 벡터 a + b로 계산된 결과입니다.

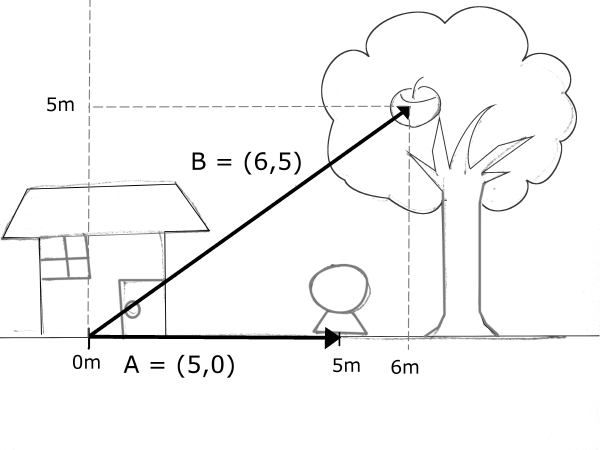
수학적으로 표현하면 아래와 같은 식이 나오죠.

a + b = [5,0] + [1,0] = [6,0]

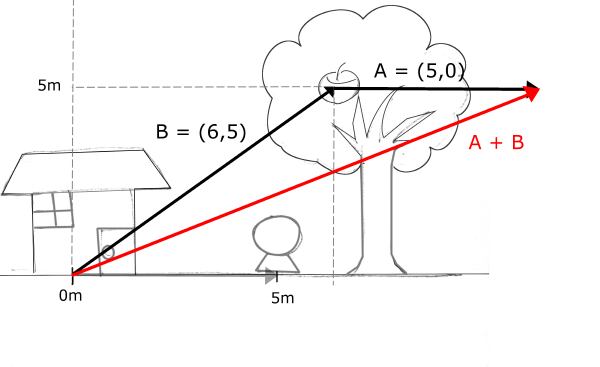
벡터의 덧셈을 그림으로 나타내면 꼬리에서 머리로 이어주는 벡터가 되며,

수식으로 나타내면 벡터의 각 요소[x,y] 끼리 더해준 결과입니다.

같은 선 위에 있지 않은 벡터도 물론 더할 수 있습니다.



위 그림에서 사람을 가리키고 있는 벡터와 사과를 가리키고 있는 벡터를 더하면 아래 그림과 같은 벡터가 생깁니다.



벡터 a를 벡터 b의 머리 부분에 이어지도록 위치 시킵니다.

벡터는 크기와 방향만 같으면 어디에 옮겨도 상관 없기 때문에

벡터 a를 b의 머리 부분으로 이동시켜도 이전에 사람을 가리키고 있던 벡터와 동일한 벡터로 취급할 수 있습니다.

이 상태에서 꼬리에서 머리로 이어주면 두 벡터를 더한 새로운 벡터(a+b)가 생깁니다.

물론 게임 프로그래밍에서 두 벡터를 더할 때 위와 같은 풀이 과정을 거칠 필요는 없습니다.

두 벡터의 각 요소[x,y]를 서로 더해주면 이 풀이 과정과 동일한 결과를 나타내기 때문에

산술적으로 계산만 해주면 됩니다.

참 쉽죠?

여기에는 단순히 두 수를 더해주는 계산만 필요하기 때문에 수학적으로 어려운 개념은 전혀 없습니다.

더하기만 할 줄 안다면 누구든지 벡터의 덧셈을 할 수 있다는 뜻이기도 하죠.

게다가 계산은 컴퓨터가 해주기 때문에 게임에 활용하는 것은 더욱더 쉬워집니다.

벡터의 덧셈 정리

1. 두 벡터끼리 서로 이어지도록 만든다.

2. 한 벡터의 꼬리에서 다른 벡터의 머리까지 이어서 새로운 벡터를 만든다.

3. 새로 만들어진 벡터가 두 벡터를 더한 것이 된다.

다음 강좌에서는 벡터의 뺄셈과 곱셈에 대해서 알아보겠습니다.